Knoxification of CDH 6.1 clusterz

Goal:
Integrate CDH 6.1 Services with Knox

Prerequisites:

- A CDH 6.1 cluster
- Kerberos enabled against FreeIPA
- Install Knox on a CDH "Edge Node" so it has the proper client configs

Knox - CDH 6.1 Service Matrix:

Service Endpoint	Knox Proxy URL	Knox SSO URL
HDFS NameNode UI	??	http://cdh-1.evilcorp.pro:9870/index.html
WebHDFS REST API	https://cdh-3.evilcorp.pro:8443/gateway/fortknox/webhdfs/v1/?op=LISTSTATUS	??
YARN Resource Manager UI	https://cdh-3.evilcorp.pro:8443/gateway/fortknox/yarn/	http://cdh-1.evilcorp.pro:8088
Job History Server	N/A	http://cdh-1.evilcorp.pro:19888/jobhistory
Spark History Server	https://cdh-3.evilcorp.pro:8443/gateway/fortknox/sparkhistory	http://cdh-1.evilcorp.pro:18088/

 # 1. Download knox server 1.2 binary

```
mkdir /opt/knox
cd /opt/knox
sudo chown noobie /opt/knox
curl -o knox-1.2.0.zip http://apache.cs.utah.edu/knox/1.2.0/knox-1.2.0.zip

sudo yum install unzip
```

```
unzip knox-1.2.0.zip
```

 # 2. Update knox configurations

```
# Inside the  /opt/knox/knox-1.2.0/conf/

# 2.1 create krb5JAASLogin.conf
com.sun.security.jgss.initiate {
    com.sun.security.auth.module.Krb5LoginModule required
    renewTGT=false
    doNotPrompt=true
    useKeyTab=true
    keyTab="/etc/security/knox.keytab"
    principal="knox/cdh-3.evilcorp.pro@EVILCORP.PRO"
    isInitiator=true
    storeKey=true
    useTicketCache=false
    client=true;
};

# 2.2    /opt/knox/knox-1.2.0/conf/gateway-site.xml

# Modify gateway-site.xml. Match /pathtoknox/conf/krb5JAASLogin.conf to the location you untarred knox :

   <property>
        <name>gateway.hadoop.kerberos.secured</name>
        <value>true</value>
        <description>Boolean flag indicating whether the Hadoop cluster protected by Gateway is secured with Kerberos</description>
    </property>
    <property>
        <name>java.security.krb5.conf</name>
        <value>/etc/krb5.conf</value>
        <description>Absolute path to krb5.conf file</description>
    </property>
    <property>
        <name>java.security.auth.login.config</name>
        <value>/opt/knox/knox-1.2.0/conf/krb5JAASLogin.conf</value>
        <description>Absolute path to JAAS login config file</description>
    </property>
```

```
# Create knox master secret
```

```
cd /opt/knox/knox-1.2.0/bin/
bin/knoxcli.sh create-master

Enter master secret: useabetterpassw0rd
```

```
# test case
```

```
curl -vv --negotiate -u: 'http://cdh-1.evilcorp.pro:9870/webhdfs/v1/?op=LISTSTATUS'


curl -i -k -u noobie:noobiepassw0rd -H "X-Requested-By: jonsnow" 'https://cdh-3.evilcorp.pro:8443/gateway/fortknox/webhdfs/v1/?op=LISTSTATUS'
```

 # 3 Give Knox a kerberos keytab

add service account in ipa

```
ipa user-add knox --first=knox --last=service-account
```

create keytab for the user

```
ipa service-add knox/cdh-3.evilcorp.pro@EVILCORP.PRO

ipa-getkeytab -p  knox/cdh-3.evilcorp.pro@EVILCORP.PRO -k /etc/security/knox.keytab -s ipa.evilcorp.pro
```

```
chown knox:hadoop /etc/security/knox.keytab
```

```
ls -lrt /etc/security/knox.keytab
-rw-------. 1 knox hadoop 176 Feb  9 01:51 /etc/security/knox.keytab

klist -kt /etc/security/knox.keytab
Keytab name: FILE:/etc/security/knox.keytab
KVNO Timestamp           Principal
---- ------------------- ------------------------------------------------------
   1 02/09/2019 01:51:03 knox/cdh-3.evilcorp.pro@EVILCORP.PRO (aes256-cts-hmac-sha1-96)
   1 02/09/2019 01:51:03 knox/cdh-3.evilcorp.pro@EVILCORP.PRO (aes128-cts-hmac-sha1-96)
```

create topology called 'fortknox' for proxy services

```
vim /opt/knox/knox-1.2.0/conf/topologies/fortknox.xml
```

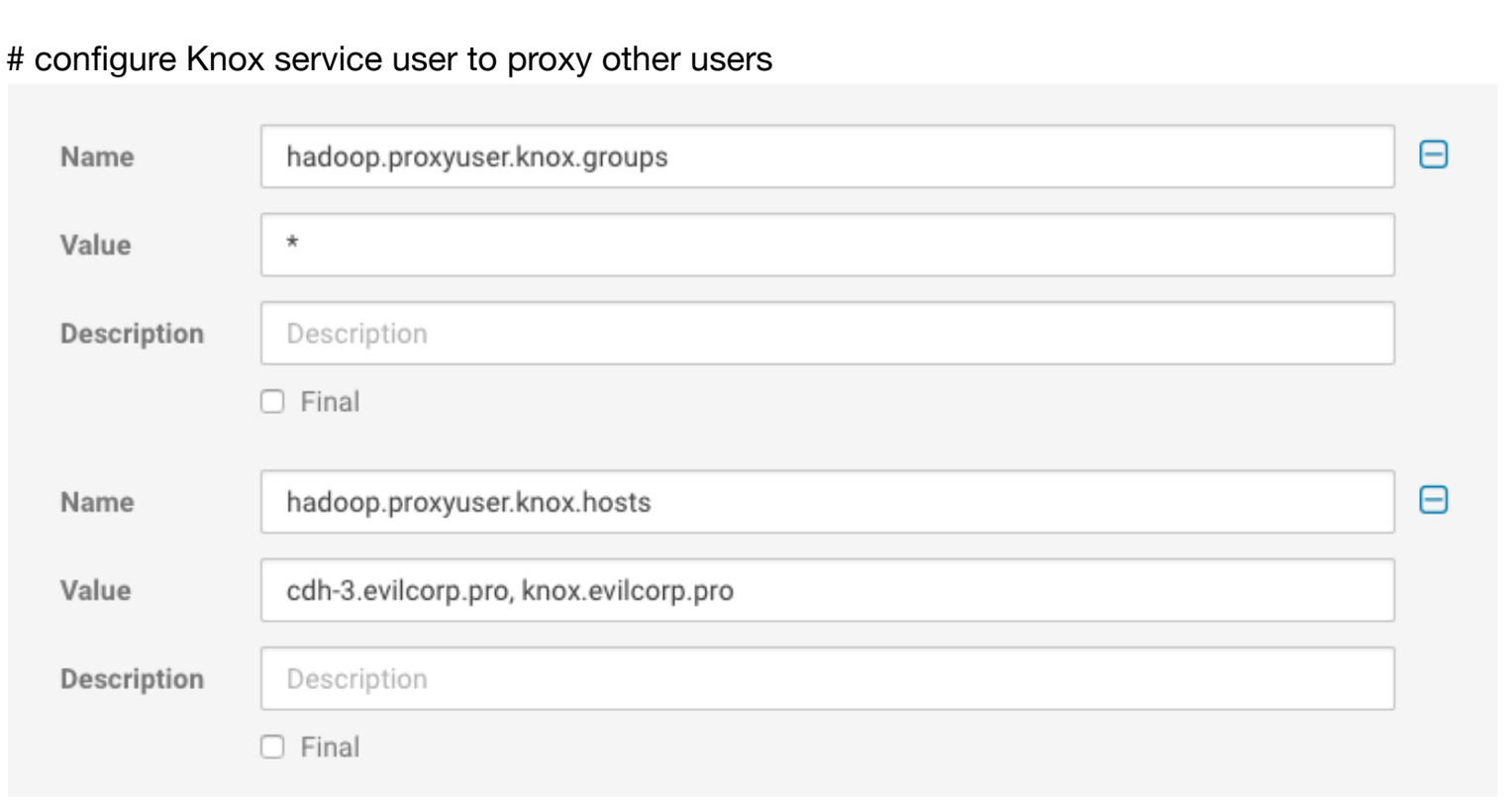
configure Knox with ldap

```
                 <param>
                     <name>main.ldapRealm.userDnTemplate</name>
                     <value>uid={0},cn=users,cn=accounts,dc=evilcorp,dc=pro</value>
                 </param>
                 <param>
                     <name>main.ldapRealm.contextFactory.url</name>
                     <value>ldaps://ipa.evilcorp.pro:636</value>
                 </param>
```

make ldaps happy for Knox

```
storepass="changeit"
JAVA_HOME=/usr/java/jdk1.8.0_141-cloudera
export PATH=$JAVA_HOME/bin:$PATH
keytool -importcert -trustcacerts -file /etc/ipa/ca.crt -storepass ${storepass} -noprompt -alias FreeIPACA -keystore ${JAVA_HOME}/jre/lib/security/cacerts
```

configure Knox service user to proxy other users




```
hadoop.proxyuser.knox.groups
hadoop.proxyuser.knox.hosts
```

change ownership of installation directory to Knox

```
chown -R knox:knox /opt/knox/knox-1.2.0
```

start Knox

```
sudo -u knox /opt/knox/knox-1.2.0/bin/gateway.sh start
```

 # 4 Proxy CDH Services via Knox

add service definitions in fortknox.xml

```
    <service>
        <role>WEBHDFS</role>
        <url>http://cdh-1.evilcorp.pro:9870/webhdfs</url>
    </service>

   <service>
      <role>YARN</role>
      <url>http://cdh-1.evilcorp.pro:8088</url>
   </service>

   <service>
      <role>YARNUI</role>
      <url>http://cdh-1.evilcorp.pro:8088</url>
   </service>

    <service>
        <role>SPARKHISTORYUI</role>
        <url>http://cdh-1.evilcorp.pro:18088/</url>
    </service>
```

Define our domain name “evilcorp.pro” in Knox gateway whitelist

```
vim /opt/knox/knox-1.2.0/conf/gateway-site.xml

    <property>
        <name>gateway.dispatch.whitelist</name>
        <value>^https?:\/\/(localhost|127\.0\.0\.1|0:0:0:0:0:0:0:1|::1|.*\.compute-1\.amazonaws\.com|.*\.evilcorp\.pro):[0-9].*$</value>
        <description>The whitelist to be applied for dispatches associated with the service roles specified by gateway.dispatch.whitelist.services.
        If the value is DEFAULT, a domain-based whitelist will be derived from the Knox host.</description>
    </property>
```

Force Knox SSO tokens to be able to send over HTTP (This is required because our CDH endpoints are not HTTPS) - NOT recommended for production as JWT cookies are now being sent over insecure channel)

```
vim /opt/knox/knox-1.2.0/conf/topologies/knoxsso.xml

        <param>
            <name>knoxsso.cookie.secure.only</name>
            <value>false</value>
        </param>
```

create home directory in HDFS

```
[root@cdh-3 noobie]# hadoop fs -mkdir /user/noobie
[root@cdh-3 noobie]# hadoop fs -chown noobie:noobie /user/noobie
```

woot, WEBHDFS works!
https://cdh-3.evilcorp.pro:8443/gateway/fortknox/webhdfs/v1/?op=LISTSTATUS
OR

```
curl -i -k -u noobie:noobiepassw0rd -H "X-Requested-By: jonsnow" 'https://cdh-3.evilcorp.pro:8443/gateway/fortknox/webhdfs/v1/?op=LISTSTATUS'
```

 # 5 Configure CDH services for SSO
Configure Knox SSO for CDH services
Apache Knox Reference Doc: https://knox.apache.org/books/knox-0-12-0/user-guide.html#Participating+Application+Configuration

---

 # 6 Replace Knox's default SSL certificate with Corporate CA-signed certificate

(Password for private key: useabetterpassw0rd)

☑ generate a new key

https://bgstack15.wordpress.com/2017/05/21/generate-certificate-with-subjectaltname-attributes-in-freeipa/

```
openssl genrsa -aes256 -out /etc/security/certificates/knox-san.key 2048
```

☑ generate csr for hostname knox.evilcorp.pro with SAN cdh-3.evilcorp.pro

```
vim /etc/pki/tls/openssl.cnf
[req]
req_extensions          = v3_req

[v3_req]
subjectAltName = @alt_names
extendedKeyUsage = serverAuth, clientAuth

[alt_names]
DNS.1 = cdh-3.evilcorp.pro
```

generate the csr

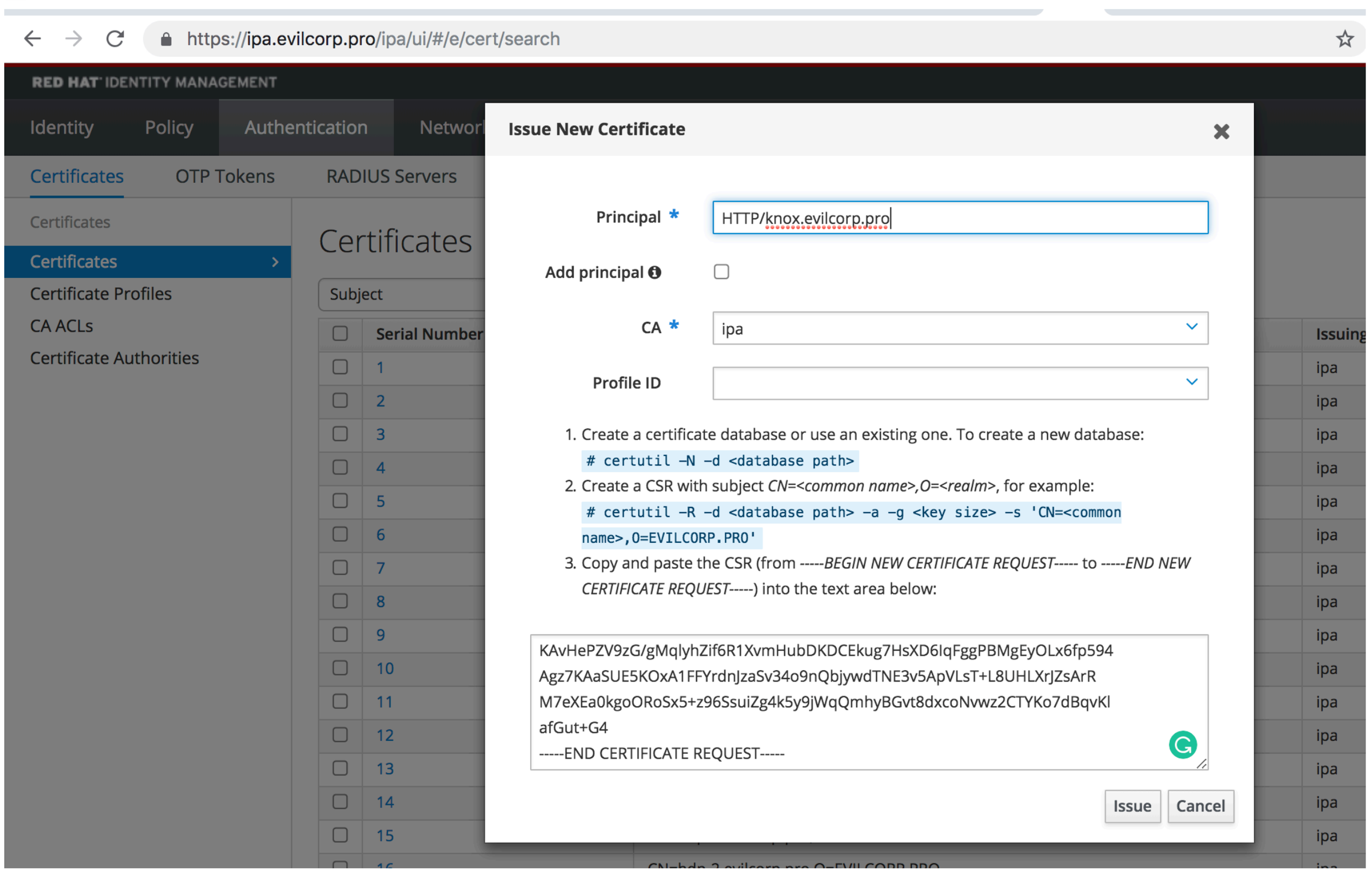
```
openssl req -new -key /etc/security/certificates/knox-san.key -out /etc/security/certificates/knox-san.csr
```

```
Enter pass phrase for /etc/security/certificates/knox-san.key:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:US
State or Province Name (full name) []:CA
Locality Name (eg, city) [Default City]:San Jose
Organization Name (eg, company) [Default Company Ltd]:Cloudera
Organizational Unit Name (eg, section) []:PM
Common Name (eg, your name or your server's hostname) []:knox.evilcorp.pro
Email Address []:youremail@evilcorp.pro

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

```
cat /etc/security/certificates/knox-san.csr
```

☑ get Knox CSR signed by Corporate CA (IPA CA in our case)



☑ replace Knox's cert on the Knox host

First thing first - Take back up and make sure yo use the same secret (useabetterpassw0rd)

```
sudo su
cd /opt/knox/knox-1.2.0/data/security/keystores/
mkdir backup2
mv gateway.jks __gateway-credentials.jceks backup2/
```

create a keystore in pkcs12 for Knox out of Knox's new private key + Signed certificate above + IPA CA cert (password for key is useabetterpassw0rd , export password is changeit)

```
openssl pkcs12 -export -out knox-san.p12 -inkey /etc/security/certificates/knox-san.key -in /etc/security/certificates/knox-san.crt -certfile /etc/ipa/ca.crt
```

find the "src alias name.." (To be used in next command) (Password is : changeit)

```
/usr/jdk64/jdk1.8.0_112/bin/keytool -list -v -keystore knox-san.p12
```

convert PKCS12 keystore into JKS keystore (gateway.jks) for Knox

```
/usr/jdk64/jdk1.8.0_112/bin/keytool -importkeystore -srckeystore knox-san.p12 -srcstoretype pkcs12 -destkeystore gateway.jks -deststoretype jks -srcstorepass changeit -deststorepass useabetterpassw0rd -srcalias 1 -destalias gateway-identity -destkeypass useabetterpassw0rd
```

We don't need to change gateway-identity-passphrase if we kept password of private key same as the master password. If not, then use the command below to change the Knox master key password.
store the keystore password in the jks file

```
/usr/hdp/current/knox-server/bin/knoxcli.sh create-alias gateway-identity-passphrase --value useabetterpassw0rd
```

change the certificate for **knox on cdh-3 host**

```
 cp /home/noobie/gateway.jks /opt/knox/knox-1.2.0/data/security/keystores/
```

restart knox

```
sudo -u knox /opt/knox/knox-1.2.0/bin/gateway.sh stop
sudo -u knox /opt/knox/knox-1.2.0/bin/gateway.sh start
```

export the Knox public certificate (to be used while setting up SSO) (from Knox on cdh-3)

```
# JAVA_HOME=/usr/java/jdk1.8.0_141-cloudera
# $JAVA_HOME/bin/keytool -export -alias gateway-identity -rfc -file /etc/security/sso.pem -keystore /opt/knox/knox-1.2.0/data/security/keystores/gateway.jks
```

`[root@knox keystores]# cat /etc/security/sso.pem`


```
-----BEGIN CERTIFICATE-----
MIIERTCCAy2gAwIBAgIBFTANBgkqhkiG9w0BAQsFADA3MRUwEwYDVQQKDAxFVklM
Q09SUC5QUk8xHjAcBgNVBAMMFUNlcnRpZmljYXRlIEF1dGhvcml0eTAeFw0xOTAy
MTMwNTI1NDlaFw0yMTAyMTMwNTI1NDlaMDMxFTATBgNVBAoMDEVWSUxDT1JQLlBS
TzEaMBgGA1UEAwwRa25veC5ldmlsY29ycC5wcm8wggEiMA0GCSqGSIb3DQEBAQUA
A4IBDwAwggEKAoIBAQDNGQmGUSDw4gdDxlA42x9u99RvJy397tmYHlpKL2q19iR/
Df/hMmBtgG483U6WB3oVO+KQrZNRj0l+VzNhgvjZg/+cXd4baRtpsk6z+by0E9u8
J+JM6geSOjXEHpbpScq/Hjyx5j7quQdoJ0smhMAB+tM08T+9ZH1yzmRpEEzb0h7x
74FOsnB9wZpPeELu/nK7k1SRkq8ldNeIeSllmuHLhEdhWoXoXbY94rp4AfLQSBdH
AqU/xLZPDl/UvC/+qriG7kbG4nm9BFXLZCZrTqsi/zwlhsarnRHU/ndUNQxiwZW8
ZRWqUPtpsR93wSVfkyqHqdwCGKQ2sWUob0rSPCRHAgMBAAGjggFeMIIBWjAfBgNV
HSMEGDAWgBQnMV3FMpBTRTRQk9TX2C6qkfiYaDA+BggrBgEFBQcBAQQyMDAwLgYI
KwYBBQUHMAGGImh0dHA6Ly9pcGEtY2EuZXZpbGNvcnAucHJvL2NhL29jc3AwDgYD
VR0PAQH/BAQDAgTwMB0GA1UdJQQWMBQGCCsGAQUFBwMBBggrBgEFBQcDAjB3BgNV
HR8EcDBuMGygNKAyhjBodHRwOi8vaXBhLWNhLmV2aWxjb3JwLnByby9pcGEvY3Js
L01hc3RlckNSTC5iaW6iNKQyMDAxDjAMBgNVBAoMBWlwYWNhMR4wHAYDVQQDDBVD
ZXJ0aWZpY2F0ZSBBdXRob3JpdHkwHQYDVR0OBBYEFLeplEZzlOQlJg0QPJ1YScw+
8HUXMDAGA1UdEQQpMCeCEmNkaC0zLmV2aWxjb3JwLnByb4IRa25veC5ldmlsY29y
cC5wcm8wDQYJKoZIhvcNAQELBQADggEBAKokBRYDkAQLnMGrtdOj31K4hg0m6Sns
DYhdOClteznM7awZCyM6JJ7vl8e8e9RDmh8kGo1mOg64QcdANwntjYDbY0nQnVci
iDVrCLHjzurxLhh2CeHdKgpQGKapm7+5jrjhECdJ/W4UV8c9WBWCUkniWcIl8Ynk
Rz390lWBzNfOSR4+Vyn41VNm2L+IUF4o+TVcK7Lr7MEJiTjx9rv5zL3NWomljDk6
6pew3FThm2XQZj3rK5dIQ4ZuVOlbSO33PVaviN5snv+DHYTvsRGayk+35IQGvSPl
0/Esq+/hNNwHqb7l9ldEDNZh3BTrd6HueqRED0DyIS4ETDIvUbuQzX8=
-----END CERTIFICATE-----
```

Configuring Knox SSO for Hadoop Core Services in Cloudera Manager

Check box the Hadoop HTTP Authentication textbox in Cloudera Manager for BOTH HDFS and YARN !!!

add custom properties in hdfs core-site safety valve

```
<property>
    <name>hadoop.http.authentication.type</name
 <value>org.apache.hadoop.security.authentication.server.JWTRedirectAuthenticationHandler</value>
</property>
```

add the Knox SSO public certificate

```
<property>
    <name>hadoop.http.authentication.public.key.pem</name>
    <value>
MIIERTCCAy2gAwIBAgIBFTANBgkqhkiG9w0BAQsFADA3MRUwEwYDVQQKDAxFVklM
Q09SUC5QUk8xHjAcBgNVBAMMFUNlcnRpZmljYXRlIEF1dGhvcml0eTAeFw0xOTAy
MTMwNTI1NDlaFw0yMTAyMTMwNTI1NDlaMDMxFTATBgNVBAoMDEVWSUxDT1JQLlBS
TzEaMBgGA1UEAwwRa25veC5ldmlsY29ycC5wcm8wggEiMA0GCSqGSIb3DQEBAQUA
A4IBDwAwggEKAoIBAQDNGQmGUSDw4gdDxlA42x9u99RvJy397tmYHlpKL2q19iR/
Df/hMmBtgG483U6WB3oVO+KQrZNRj0l+VzNhgvjZg/+cXd4baRtpsk6z+by0E9u8
J+JM6geSOjXEHpbpScq/Hjyx5j7quQdoJ0smhMAB+tM08T+9ZH1yzmRpEEzb0h7x
74FOsnB9wZpPeELu/nK7k1SRkq8ldNeIeSllmuHLhEdhWoXoXbY94rp4AfLQSBdH
AqU/xLZPDl/UvC/+qriG7kbG4nm9BFXLZCZrTqsi/zwlhsarnRHU/ndUNQxiwZW8
ZRWqUPtpsR93wSVfkyqHqdwCGKQ2sWUob0rSPCRHAgMBAAGjggFeMIIBWjAfBgNV
HSMEGDAWgBQnMV3FMpBTRTRQk9TX2C6qkfiYaDA+BggrBgEFBQcBAQQyMDAwLgYI
KwYBBQUHMAGGImh0dHA6Ly9pcGEtY2EuZXZpbGNvcnAucHJvL2NhL29jc3AwDgYD
VR0PAQH/BAQDAgTwMB0GA1UdJQQWMBQGCCsGAQUFBwMBBggrBgEFBQcDAjB3BgNV
HR8EcDBuMGygNKAyhjBodHRwOi8vaXBhLWNhLmV2aWxjb3JwLnByby9pcGEvY3Js
L01hc3RlckNSTC5iaW6iNKQyMDAxDjAMBgNVBAoMBWlwYWNhMR4wHAYDVQQDDBVD
ZXJ0aWZpY2F0ZSBBdXRob3JpdHkwHQYDVR0OBBYEFLeplEZzlOQlJg0QPJ1YScw+
8HUXMDAGA1UdEQQpMCeCEmNkaC0zLmV2aWxjb3JwLnByb4IRa25veC5ldmlsY29y
cC5wcm8wDQYJKoZIhvcNAQELBQADggEBAKokBRYDkAQLnMGrtdOj31K4hg0m6Sns
DYhdOClteznM7awZCyM6JJ7vl8e8e9RDmh8kGo1mOg64QcdANwntjYDbY0nQnVci
iDVrCLHjzurxLhh2CeHdKgpQGKapm7+5jrjhECdJ/W4UV8c9WBWCUkniWcIl8Ynk
Rz390lWBzNfOSR4+Vyn41VNm2L+IUF4o+TVcK7Lr7MEJiTjx9rv5zL3NWomljDk6
6pew3FThm2XQZj3rK5dIQ4ZuVOlbSO33PVaviN5snv+DHYTvsRGayk+35IQGvSPl
0/Esq+/hNNwHqb7l9ldEDNZh3BTrd6HueqRED0DyIS4ETDIvUbuQzX8=
</value>
</property>
```

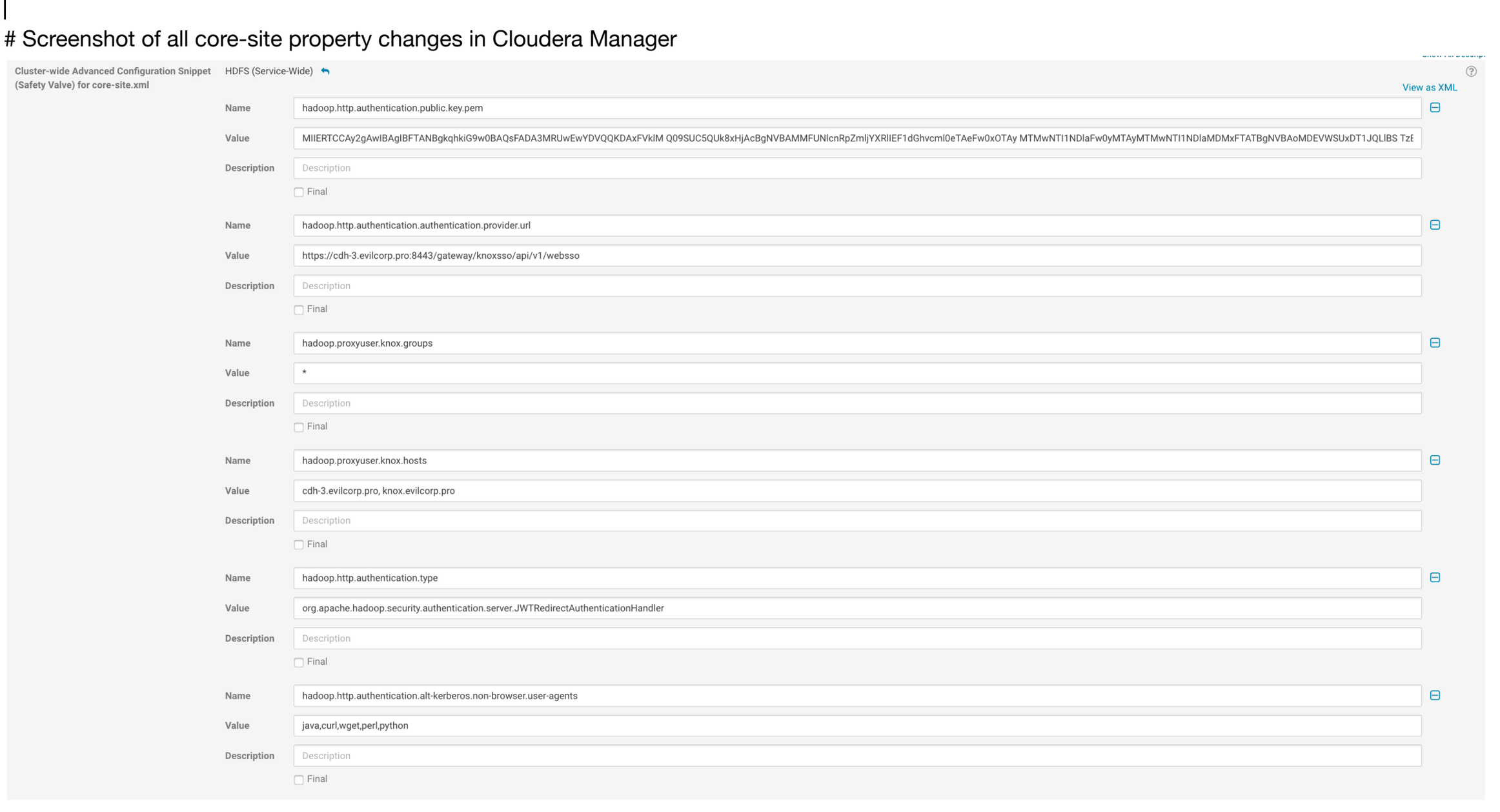
Add Knox SSO provider URL

```
<property>
    <name>hadoop.http.authentication.authentication.provider.url</name>
    <value>https://cdh-3.evilcorp.pro:8443/gateway/knoxsso/api/v1/websso</value>
</property>
```

Add white listing for user-agents where SSO is NOT applicable (they fallback to Kerberos authentication)

```
<property>
    <name>hadoop.http.authentication.alt-kerberos.non-browser.user-agents</name>
    <value>java,curl,wget,perl,python</value>
</property>
```

Screenshot of all core-site property changes in Cloudera Manager



Config changes for Spark History Server in safety valve (CDH cluster)

```
# Spark Service Environment Advanced Configuration Snippet (Safety Valve)
ENABLE_SPNEGO=false
YARN_PROXY_REDIRECT=false

# History Server Advanced Configuration Snippet (Safety Valve) for spark-conf/spark-history-server.conf
spark.history.ui.acls.enable=true
spark.ui.filters=org.apache.spark.deploy.yarn.YarnProxyRedirectFilter,org.apache.hadoop.security.authentication.server.AuthenticationFilter
spark.org.apache.hadoop.security.authentication.server.AuthenticationFilter.param.type=org.apache.hadoop.security.authentication.server.JWTRedirectAuthenticationHandler
spark.org.apache.hadoop.security.authentication.server.AuthenticationFilter.param.kerberos.principal=HTTP/cdh-1.evilcorp.pro@EVILCORP.PRO
spark.org.apache.hadoop.security.authentication.server.AuthenticationFilter.param.kerberos.keytab=spark_on_yarn.keytab
spark.org.apache.hadoop.security.authentication.server.AuthenticationFilter.param.kerberos.name.rules=DEFAULT\u000A
spark.org.apache.hadoop.security.authentication.server.AuthenticationFilter.param.authentication.provider.url=https://cdh-3.evilcorp.pro:8443/gateway/knoxsso/api/v1/websso
spark.org.apache.hadoop.security.authentication.server.AuthenticationFilter.param.public.key.pem=MIIERTCCAy2gAwIBAgIBFTANBgkqhkiG9w0BAQsFADA3MRUwEwYDVQQKDAxFVklMQ09SUC5QUk8xHjAcBgNV
BAMMFUNlcnRpZmljYXRlIEF1dGhvcml0eTAeFw0xOTAyMTMwNTI1NDlaFw0yMTAyMTMwNTI1NDlaMDMxFTATBgNVBAoMDEVWSUxDT1JQLlBSTzEaMBgGA1UEAwwRa25veC5ldmlsY29ycC5wcm8wggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwg
gEKAoIBAQDNGQmGUSDw4gdDxlA42x9u99RvJy397tmYHlpKL2q19iR/Df/hMmBtgG483U6WB3oVO+KQrZNRj0l+VzNhgvjZg/+cXd4baRtpsk6z+by0E9u8J+JM6geSOjXEHpbpScq/Hjyx5j7quQdoJ0smhMAB+tM08T+9ZH1yzmRpEEzb0h
7x74FOsnB9wZpPeELu/nK7k1SRkq8ldNeIeSllmuHLhEdhWoXoXbY94rp4AfLQSBdHAqU/xLZPDl/UvC/+qriG7kbG4nm9BFXLZCZrTqsi/zwlhsarnRHU/ndUNQxiwZW8ZRWqUPtpsR93wSVfkyqHqdwCGKQ2sWUob0rSPCRHAgMBAAGjggF
eMIIBWjAfBgNVHSMEGDAWgBQnMV3FMpBTRTRQk9TX2C6qkfiYaDA+BggrBgEFBQcBAQQyMDAwLgYIKwYBBQUHMAGGImh0dHA6Ly9pcGEtY2EuZXZpbGNvcnAucHJvL2NhL29jc3AwDgYDVR0PAQH/BAQDAgTwMB0GA1UdJQQWMBQGCCsGAQUF
BwMBBggrBgEFBQcDAjB3BgNVHR8EcDBuMGygNKAyhjBodHRwOi8vaXBhLWNhLmV2aWxjb3JwLnByby9pcGEvY3JsL01hc3RlckNSTC5iaW6iNKQyMDAxDjAMBgNVBAoMBWlwYWNhMR4wHAYDVQQDDBVDZXJ0aWZpY2F0ZSBBdXRob3JpdHkwH
QYDVR0OBBYEFLeplEZzlOQlJg0QPJ1YScw+8HUXMDAGA1UdEQQpMCeCEmNkaC0zLmV2aWxjb3JwLnByb4IRa25veC5ldmlsY29ycC5wcm8wDQYJKoZIhvcNAQELBQADggEBAKokBRYDkAQLnMGrtdOj31K4hg0m6SnsDYhdOClteznM7awZCy
M6JJ7vl8e8e9RDmh8kGo1mOg64QcdANwntjYDbY0nQnVciiDVrCLHjzurxLhh2CeHdKgpQGKapm7+5jrjhECdJ/W4UV8c9WBWCUkniWcIl8YnkRz390lWBzNfOSR4+Vyn41VNm2L+IUF4o+TVcK7Lr7MEJiTjx9rv5zL3NWomljDk66pew3FT
hm2XQZj3rK5dIQ4ZuVOlbSO33PVaviN5snv+DHYTvsRGayk+35IQGvSPl0/Esq+/hNNwHqb7l9ldEDNZh3BTrd6HueqRED0DyIS4ETDIvUbuQzX8=
```

 # Cloudera Manager Tips for using Knox:

- Namenode UI link - Need to use http://cdh-1.evilcorp.pro:9870/index.html and NOT http://ec2-52-206-223-66.compute-1.amazonaws.com:9870/ to ensure that the cookie domains match. CM currently has AWS domain name for quick link
- Make sure to configure http auth for HDFS & YARN!!